

محاضرة 5 : أساسيات Digital Image - تابع العمليات الرياضية

المحتوى : ① Spatial operations and transformation

Image Registration ②

Image Transform ③

Probabilistic Models ④

Spatial operations and transformation
* فيه نوعين من ال Transformation ، واحد في ال Spatial domain (إحداثيات x, y)

وفيه راند حكم أعد ال Transform للصورة بحيث أنقلها لـ Domain ثاني زي ال Z-domain أو ال frequency domain (معرض عن ال pixel للصورة لـ Z-domain بس حكم يعني)

* بشكل عام ، حكم تقسم ال operations على الصور كالتالي :

① Spatial Operations

* قولنا بتعمل في ال Spatial domain بإحداثيات x, y حيث

$x = 0, 1, 2, 3, \dots, M$ و $y = 0, 1, 2, 3, \dots, N$

* العمليات حكم تكون عمليات على البكسل لو مده (Single pixel based operation)

أو عمليات على ال Pixels المجاورة (neighborhood operations)

أو عمليات Geometric Transformation زي ال rotation و ال Translation و ال Shearing .

② Transform Operations

* بنقل الصورة لـ Domain ثاني زي ال frequency domain

* بعمل اللي عايزه على الصورة في ال Domain الجديد

* بعدها أعد ال Inverse Transform عتس n أربحها لـ Spatial domain

العمليات في ال Spatial Domain

① Single pixel Based

أجد العمليات هي ال intensity mapping

* فليها نقول الصورة ال S عملية هي T ، والصورة بعد التحويل هي S وال عملية

الها T ، حكم نكتبها على الصورة

$$S = T(z)$$

* لو قلت مثلا دانه عندي صورة عايز أجيپ ال Negative ليها لايحي
أطرح كل الأرقام من أكبر قيمة ممكنة ال pixel اللي هي $2^8 - 1 = 255$

الصورة بعد ال Negation operation هتبقى S
الصورة الأصلية هي T ، العملية هتبقى T

$$S = T(z) = (2^8 - 1) - z = 255 - z$$

* معنى المعادلة داني همشتي على كل بكسل في الصورة T وأطرحه من 255
عشانه يديني الصورة S

Neighborhood Spatial operations (2)

* من التطبيقات عليها هي عملية ال Blur

* لو فرضت بكسل P ، عندي مجموعة

neighbors من $n_1 \rightarrow n_{14}$ هتخلي

قيمة ال P كالتالي

$$P_{\text{new}} = \frac{n_1 + n_2 + n_3 + \dots + n_{14} + P}{\text{مجموع البكسلز}}$$

عدد البكسلز $M \times N$

* لو عايز أعمم الكلام دة على صورة $f(r, c)$ حيث r, c هي الإحداثيات

لكل بكسل :

- أولا ، بافديني من ال $M \times N$ حيث اننا في الالة دي بتغير أنا باخذ

Neighbors قدرناك لو خلت ال pixel في النصف زي ال رسمه فوق (اد Window) بتاعتي

- ثانيا : ممكنه نعيد كتابة الحساب موف

$$P_{\text{new}} = \frac{1}{M \times N} \sum_{i=1}^{M \times N} n_i$$

ده لو فرضنا بكسل واحد بس

- نعم الكلام ده للصورة $f(r, c)$ الأصلية ، عشان أجيپ عنو $g(x, y)$
بعد ال blur

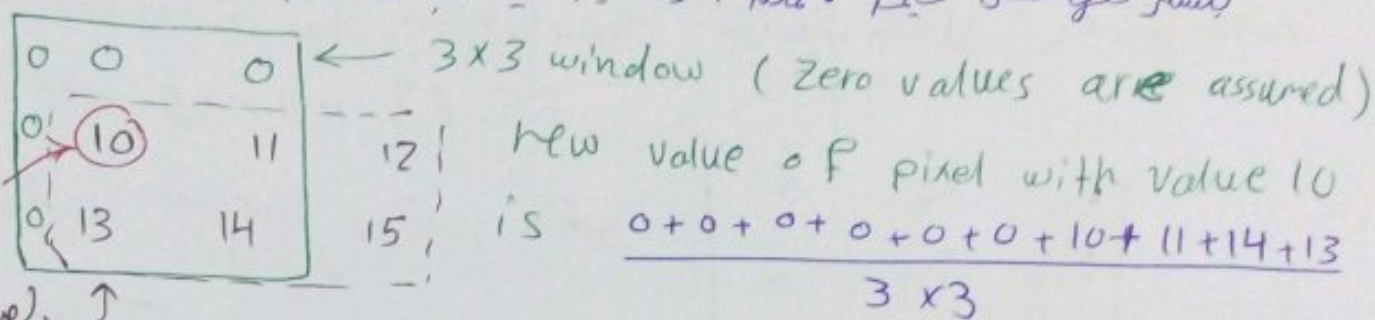
$$g(x, y) = \frac{1}{M \times N} \sum_{(r, c) \in S_{x, y}} f(r, c)$$

معنى المعادلة ، انه لكل بكسل عايز أجيپه في $g(x, y)$ هجيپ ال window

بجيم $M \times N$ على ال pixel الياظر في $f(r, c)$ وأجيپ قيمته

زي ما علمت في P_{new}

* ملاحظة ، لم عندي بكسل في أطراف الصورة هلاقي اد window حافية بكسلاتها مش قيمة ، فتم افترض القيمة دي بصفر أو بأقرب قيمة مجاورة مثلاً



البكسل
التي تعمل عليه
العملية

↑
صورة الإدخال

Geometric spatial Transformation (3)

الجزء ده مشروح كويس في صفحة 22 وحلايد [5.5] ، هنوضح شوية

ملاحظات وتفهم اد Affine Transformation

* اعمل معايا skip لسلايد [5.9] صراحي الصورة (a) اتعملها Shearing وقت (b)

- مشكلة اد holes هي اننا كمان الي بقى بعد اد shearing

- مشكلة اد overlap هي راند مع اد shearing بقى فيه pixels في الصورة ممكن يكون لها أكثر من قيمة نزي اد pixels في انزياح.

* طب لو فرضت راند الصورة (a) كانت الصورة الأصلية ، وعاليز أزبط الصورة دي أفليها زي (a) ثاني ، هيبقى عندي مشكلة إنده فيه قيم بكسلز كانت أصلاً بصفر أو قيمتها مش معلومة (الحدود السوداء في (b) وعاليز أجيب قيمتها .

- في الحالة دي هنعمل interpolation بأنواعه الثلاثة زي ما قولنا قبل كدة في محاضرة 3

* نرتمه اد interpolation لها مثال ثاني في حلايد [5.7]

- هلاقي مع اد rotation بيبقى عندي jagged ، عاليز أعملها Smoothing

ومببهاش التغير jagged ، فنبعمل Edges interpolation ، هلاقي

التغير بقى فيه تدرج في قيم اد pixels مش حادة بحيث يقل اد jagged edges

* نتكلم بقى عن اد Affine transformation

* عندي عملياً كتير ممكن أعملها زي اد scaling وار rotation

وار Translation وار shearing

* طاقا الصورة أحذر أمثلها في شكل 2D-Array أو Matrix ، ليك مجتمش العمليات

دي في صنف واحد أو غير في القيم [3] عشانه أعمل العملية الي عاليزها ؟

* لو خاكر في الهندسة التحليلية في إيمداري، كفتت تقدر تعمل عمليات
 rotation و Translation مع ضلاد معادلات، شكل المعادلة
 لكل عملية في سلايد [5.6]

$$\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

* شكل ال Affine Transform هو Matrix

* ونسميها T

* بتغير قيم ال z في t ممكنة أعمل على عمليات
 التي محتاجها.

* لو فرضت عندك مجموعة بكسز، كل بكسل نرسل للإحداثيات بتاعته بـ (u, v)
 "بدا x, y" و بعد ما حولت ال pixel بقى إحداثياته (u', v') أوزي

ماهي مكتوبة في المحاضرة (x, y) ممكن نقول

الإحداثيات بعد
 Transformation

$$(x, y) = T \{ (u, v) \}$$

* هدفنا صياغة المعادلة في شكل vectors و Matrices عشانه نستخدم
 عملية ال Transform

ال Affine transformation Matrix

$$[x, y, 1] = [u, v, 1] \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

* لما نيجي نطبق الكلام ده على المعادلات في سلايد [5.6]

* هنطبق على ال Translation. لو عاينز ال pixels كلها بمقدار 4 على محور y و 5 على محور x

* هنفرض إننا الإحداثيات الأصلية كانت (u, v)

و الإحداثيات بعد التحويل هي (u', v')

* تطبيق التحويل كالتالي

$$[v', w', 1] = [v, w, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 4 & 1 \end{bmatrix}$$

$$= [v(1) + (0)w + 5, (0)v + (1)w + 4, (0)v + (0)w + 1]$$

$$= [v + 5, w + 4, 1]$$

$$\therefore v' = v + 5, w' = w + 4$$

* امشي بقية المعادلات بنفس الطريقة
* ما احب ارجع للأصل بعمل T^{-1} : inverse Transform

Image Registration
مشرحة كويس في صفحة 23 و سلايد [5.8, 5.9]

* اقرؤوا سلايد [5.10] و [5.11]
* فكرة المعادلة في [5.10] انه بدل ما كان بيحسب الـ Euclidean distance
للـ grayscale image و بيحسب لـ صورة RGB ، فبيعامل مع كل لون لونه و يجمعهم
* يقدر يكتبها في صورة Matrix
- سلايد [5.11] حش فاهمها ، و افكر الدكتور مدقق فيها أوي

Image Transform
نفرق بين الـ Transform الـ Domain ثاني (Fourier transform)
والـ Geometric Transformation في الـ Spatial Domain زي الـ Rotation
* قولنا في الـ Transform نقل الصورة لـ Domain ثاني و أكمل عليها عملية
في الـ Domain الجديد بعد ما جعها لـ Spatial Domain
بالـ inverse Transform

عشانه اعداد Transform ، بطبقه معادله اسمها از Kernel
كل پیکسل بچوله $(f(x,y))$ بپيچولاي $T(u,v)$ (مجرد اسم)

ال Kernel عشانه اعمل از Transform بيرزلهاب $r(x,y,u,v)$

ال inverse transform بيرزلهاب $s(x,y,u,v)$
ال Kernel بقاخذ مجموعه اعداديات و قولها لاعدائيات جديدة ، عشانه كده
ببقى دالاه في x,y,u,v

عشانه اصب ال pixel الجديد ، لازم لجسبك بقطبيجك ال Kernel على كل
ال pixels الاصلية

$$\overset{\substack{\text{بپيچل} \\ \text{الجديد}}}{T}(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) r(x,y,u,v)$$

لا اصب اعداد inverse عشانه اصب $f(x,y)$ كاني

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v) s(x,y,u,v)$$

ال بقول على ال Kernel ، ال Seperable لو قدرت افضله لاشيه kernels
كل Kernel بيقول اعدادي x او y

$$r(x,y,u,v) = r_1(x,u) r_2(y,v)$$

بقول ال Kernel ، ال Symmetric لو كان r_1, r_2 functionally equal

(متر فاهم functionally equal بس اعتقد انه يقصد لو قربت
أفقد r_1 على x ، و قربت r_2 على y برديو ، هيدون نفس النتيجة)

ال مثال على ال Transform هو ازالة ال noise في ال frequency domain
عليها مثال في سلايد 5.14 ومحاضرة [10] عليه

* ممكن نعمل ال Transform ب Matrix operations • بس لازم شوية شروط في الحالة دي: [نفس الكلام ده Transform matrix notation]
 (1) لازم ال Kernel يكون Symmetric و separable

(2) لازم الصورة تكون أبعادها مربعة $M \times M$

* لو قولنا الصورة في شكل ال Matrix هنسماها F ، و الصورة بعد التحويل اسمها T ، و هعمل ال Transform ب Matrix اسمها A هيبقى فيه

$$T = A F A \quad (1)$$

* لما أعمل اللي عايزه على الصورة وعايز أرجعها من T إلى F هسعمل

Matrix اسمها B حيث $B = A^{-1}$

نضرب المعادلة فوق في B

$$B T B = B A F A B \quad (2)$$

$$\Rightarrow F = B T B \quad (3)$$

* ممكن يحصل عندي راند B متبقاش بالزبط تساوي A^{-1} (نتيجة عمليات تقريب للأرقام مثلا) في الحالة دي ممكن نعمل Approximation .

← هنفعل معادلة (2) ونبقى بالشكل ده

$$\hat{F} = B A F A B$$

Probabilistic Models

* الفكرة من تمثيل الصورة في ال Model كأي فير المعناري 2D-Array ، انه ده ممكن سبيل عليا محليات معينة زي ال Stochastic processes ، و ممكن مثلا تكون Models

بندي أداء أوسع في تطبيقات معينة (مثل تأكيد بلس قول ال Machine learning حاجة محتملة فيه الكلام ده)

* هتستخدم مع الصورة و ال intensities فيها ك random quantities

* ال Model ده مستخدم في Histogram processing في محاضرة 7 و 8

* لو عندي intensity قيمتها z_k حيث z_k قيمتها $0 \leq z_k \leq L-1$

* في حالتنا $0 \leq z_k \leq 255$ عشان grayscale
 * احتمالية z_k هي عدد pixels اللي ظهرت بالقيمة z_k n_k كل عدد البكسلز كلها

$$P(z_k) = \frac{n_k}{MN} \quad (1) \leftarrow$$

وطبعاً عارفين إنك الاحتمالات كلها مجموعها ب 1 ، يعني احتمالات القيم من صفر إلى $L-1$ هتبقى

$$\sum_{k=0}^{L-1} P(z_k) = 1 \quad (2) \leftarrow$$

* بناء على ال Model ده ممكن أضيف شوية characteristics زي
 ال mean و ال variance وتوزيع قيم ال pixels ، ومنها أستفيد في
 تطبيق زي تحسين ال Contrast من خلال ال Histogram processing

* لو عايز أكتب ال mean للصورة (أوار average) و ال intensities
 هتجمع قيم ال sites لكد ال pixels وأقسم على عددها

* أنا عارف إنك الصورة فيها $M \times N$ من ال pixels وكل قيمة من 0-255 z_k

ليها n_k من ال pixels ، يعني $\sum_{k=0}^{L-1} n_k z_k$ هتدي مجموع قيم
 كد ال pixels في الصورة ، و يبقى ال mean

$$m = \frac{1}{MN} \sum_{k=0}^{L-1} n_k z_k \quad (3) \leftarrow$$

* من معادلة (1) ممكن نكتب

$$m = \sum_{k=0}^{L-1} P(z_k) \cdot z_k \quad (4) \leftarrow$$

* ممكن نحسب ال Variance (بعرفنا توزيع القيم وفند نفرف ال contrast)

$$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2 P(z_k)$$

* مقياس شدة الزاوية σ^2 التي هي الـ Variance عبارة عن الـ second moment

* كما يقي أشد توزيع الـ intensity values يستعمل الـ Standard deviation

و يقارن برقم معين على قدر أصف الـ Contrast على كافي ولا لا

زي سلايد 5.16 * (Standard deviation = SD)

* أرقام الـ SD تبقى أصغر وأسهل في التعامل مع الـ Variance

و بحسب المعادلة $SD = \sigma = \sqrt{\sigma^2}$ ←

* معكس نعيم الـ moment يكون n^{th} moment

$$\sigma^n = \sum_{k=0}^{L-1} (z_k - m)^n p(z_k)$$

راجع سلايد (5.17)

* بالنسبة للـ 3rd moment يعرف معظم قيم الـ intensities هل هي أكبر من

الـ mean ولا أصغر ولا متوزعة بانتظام حول الـ mean

3rd moment (+ve) \Rightarrow pixel values bias to values higher than mean (m)

3rd moment (-ve) \Rightarrow pixel values bias to values lower than mean (m)

3rd moment (zero) \Rightarrow pixel values are uniformly distributed around the mean